

# Static SQL

With static SQL, data in an SQL-based DBMS (DB2 or SQL/DS) is accessed using an application plan. Accessing data with static SQL is faster than with dynamic SQL.

Natural for DB2 supports the use of static SQL. If a Natural program uses static SQL, a DBRM (database request module) must be generated for that program. In DB2, this DBRM must be included in an application plan.

This section covers the following topics:

- General Information
  - Documenting the Use of Static SQL
  - Generating DBRMs from Predict Documentation
  - Retrieval Functions and Consistency Checking
  - Using Predict Information when Binding Application Plans
- 

## General Information

Predict supports static SQL in several ways:

- DBRMs can be generated with the Natural for DB2 function CREATE DBRM from information stored in Predict, and Predict documentation of DBRMs can be partially generated from XRef data of existing DBRMs with the program maintenance function Redocument program. Hence Predict supports implementing and documenting static SQL no matter which one of them is done first.
- Programs that use dynamic SQL instead of static SQL can be detected easily.
  - If a program that uses static SQL has been modified and recataloged, the information stored in the DBRM for that program is no longer correct. In this case, the program automatically switches back to the use of dynamic SQL. No action has to be taken by the programmer. The switch back to dynamic SQL is therefore not necessarily recognized by the user (only increasing response times might indicate that dynamic SQL is being used again).
  - Renaming a program that uses static SQL leads to an error at execution time.
- With Predict active retrieval functions, programs that have switched back to the use of dynamic SQL because of modifications or that have been renamed after DBRM generation can be found easily.

## Documenting the Use of Static SQL

DBRMs are documented with program objects with language Q (Static SQL). Static SQL is treated in Predict like a Third Generation Language (3GL).

- An implemented DBRM is referenced in a Predict program object by an 8-character member name. A member is a set of XRef data created for the DBRM (as with 3GL programs).
- With members, an 8-character 3GL library name can be specified to identify the load library of the DBRM. A library must have been documented with a Predict system object of type G (3GL Application). This system object can be used to document the load library.

If a 3GL application has not yet been documented with a system of type G, the default DBRM library \*SYSSTA\* is used.

Both the 8-character member name and the 8-character library name belong to the implementation pointer of the program object documenting the DBRM and hence connect the documentation object to the implemented DBRM.

## Documenting Which Natural Programs Use a DBRM

A DBRM is typically used by several Natural programs, which may or may not belong to the same library. Programs using the same DBRM must be stored in the same user system file.

For each Natural program using a DBRM, an entry point must be defined in the documentation of the DBRM. The procedure is as follows:

The Natural program is entered in the entry point list with the Link Editor and Predict generates a unique entry point name for this program in the entry point list. Each entry point name is concatenated from

- the DBRM library name (not if the default library \*SYSSTA\* is used)
- the DBRM member name
- the Natural library name
- the Natural member name.

The Natural-based Link Editor is available for maintaining entry point lists in Predict:

- Natural programs documented in Predict can be selected from a list. See command **SELECT** in the section **Editors in Predict** in the **Predict Reference documentation**.
- If the DBRM has already been implemented and XRef data exists, the entry points can be derived with the commands **ACTIVE** or **UPDATE**.
  - **ACTIVE** reads the entry point names from the XRef data of the DBRM into the editor workspace and marks them as < active. Entry point names that have been entered manually but are not in the XRef data are marked as < unused.
  - **UPDATE** additionally deletes the entry points marked as < unused from the editor workspace.

## Generating DBRMs from Predict Documentation

DBRMs can be generated from Predict program objects of language Q with the Natural for DB2 function **CREATE DBRM**. XRef data for the DBRM can then be created as well.

The names of Natural programs for which DBRMs are to be generated can be specified in two ways:

- Directly as input data to the Natural for DB2 function **CREATE DBRM**.
- By using the entry point list of the Predict object.  
In this case, the Predict object to be used for the DBRM creation must be specified with the following options of the **CREATE DBRM** function:
  - The option **USING PREDICT DOCUMENTATION** must be specified.
  - The 8-character member name of the DBRM  
(**CREATE DBRM** <member name> ...).
  - The 3GL library the DBRM is assigned to can be specified with the parameter **LIB** <library name>.  
Predict then searches for a DBRM documentation with the <member name> and <library name> in the implementation pointer.  
If no <library name> is specified, the default library \*SYSSTA\* is searched for the given <member name>.

In both cases, XRef data can be created (provided that all Natural members themselves have been cataloged with XRef data). The **XREF** option (N, Y, F) of the Natural for DB2 function **CREATE DBRM** determines how the generation function behaves with respect to documentation and XRef data for the DBRM:

- If the **XREF** option is N, no XRef data will be written.  
Existing XRef data will be deleted.
- If the **XREF** option is Y, XRef data will be written.  
Existing XRef data will be overwritten.

- If the XREF option is F, the DBRM generation is only executed if the DBRM has already been documented in Predict. If this is true, XRef data will be written and existing XRef data will be overwritten.

A default value for the XREF option for DBRMs can be defined (use function Defaults > General Defaults > Miscellaneous > Static SQL XREF).

This default value can be changed for a single execution of the CREATE DBRM function

- from N to Y or F, or
- from Y to F

It cannot be changed from

- from F to Y or N, or
- from Y to N.

## Which Information is Stored in XRef Data

XRef data for a DBRM contains the following information:

- which files and fields are accessed via the DBRM,
- which Natural members use the DBRM. A list of entry names of the DBRM are generated from the names of the Natural members for which the DBRM is generated. The construction of the entry names is the same as in the documentation of the DBRM.

Creation of a DBRM with the XREF option set to Y or F also affects the XRef data of the Natural programs for which the DBRM was created.

Since the Natural program now 'uses' the DBRM to access the database with Static SQL, this is noted in the XRef data of the Natural program as a CALL reference to the corresponding entry point in the DBRM (special call-type Static SQL).

## Retrieval Functions and Consistency Checking

If a Natural program using static SQL has been modified and recataloged, the DBRM must be regenerated. Otherwise the program will automatically switch back to the use of dynamic SQL. This is reflected in the XRef data written when recataloging the program: It no longer contains a CALL reference to the DBRM.

However, an unused entry point in the documentation of the DBRM remains. This indicates that a Natural program designed for the use of static SQL via a DBRM in fact uses dynamic SQL. It is therefore possible to check if DBRMs have to be regenerated by comparing the Predict program object documenting a DBRM and the corresponding DBRM member (XRef data).

The following functions are available:

- LIST XREF for 3GL functions show XRef data for DBRMs.
- The Verify consistency function of LIST XREF shows all programs which have been renamed after DBRM generation.
- The File Active Retrieval function List files accessed via dynamic SQL shows all DB2 files that are accessed by Natural without using a DBRM.
- The Member Active Retrieval function List members using dynamic SQL shows all Natural members using DB2 files without using a DBRM.

## Using Predict Information when Binding Application Plans

Natural for DB2 can use information in Predict to bind plans. See your Natural for DB2 documentation for details. Information in Predict packagelist objects is then used to build the MEMBER and the PKLIST CLAUSE of a BIND statement. The following is created depending on the type of packagelist:

Type of Packagelist	PKLIST CLAUSE or MEMBER Created
Total collection (T)	PKLIST (location.collection.*, ...)
Subcollection (S)	PKLIST (location.collection.package_ID, ...)
Member (Q)	MEMBER (DBRM_name, ...)

**Note:**

The member name of the Predict program object documenting a package is interpreted as the package\_ID or DBRM\_name.